


```
PPPPPPPP      AAAAAA      SSSSSSSS  LL      000000      000000      KK      KK      AAAAAA      HH      HH
PPPPPPPP      AAAAAA      SSSSSSSS  LL      000000      000000      KK      KK      AAAAAA      HH      HH
PP      PP      AA      AA      SS      LL      00      00      00      00      KK      KK      AA      AA      HH      HH
PP      PP      AA      AA      SS      LL      00      00      00      00      KK      KK      AA      AA      HH      HH
PP      PP      AA      AA      SS      LL      00      00      00      00      KK      KK      AA      AA      HH      HH
PP      PP      AA      AA      SS      LL      00      00      00      00      KK      KK      AA      AA      HH      HH
PPPPPPPP      AA      AA      SSSSSS  LL      00      00      00      00      KKKKKK      AA      AA      HHHHHHHHHH
PPPPPPPP      AA      AA      SSSSSS  LL      00      00      00      00      KKKKKK      AA      AA      HHHHHHHHHH
PP      AAAAAAAAAA      SS      LL      00      00      00      00      KK      KK      AAAAAAAAAA      HH      HH
PP      AAAAAAAAAA      SS      LL      00      00      00      00      KK      KK      AAAAAAAAAA      HH      HH
PP      AA      AA      SS      LL      00      00      00      00      KK      KK      AA      AA      HH      HH
PP      AA      AA      SS      LL      00      00      00      00      KK      KK      AA      AA      HH      HH
PP      AA      AA      SSSSSSSS  LLLLLLLLLL      000000      000000      KK      KK      AA      AA      HH      HH
PP      AA      AA      SSSSSSSS  LLLLLLLLLL      000000      000000      KK      KK      AA      AA      HH      HH
                                                                ....
                                                                ....
                                                                ....
                                                                ....

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```



```
1 0001 0 MODULE PAS$LOOK_AHEAD ( %TITLE 'Resolve lazy-lookahead'
2 0002 0 IDENT = '1-002' ! File: PASLOOKAH.B32 Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This procedure is called by the compiled code when a textfile
36 0036 1 file buffer is undefined. If lazy-lookahead is in progress
37 0037 1 for that file, it gets the next record and returns.
38 0038 1
39 0039 1 ENVIRONMENT: User mode - AST reentrant
40 0040 1
41 0041 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original. SBL 1-April-1981
46 0046 1 1-002 - Look at the new FCB$V INITIATE_PROMPT to see if PAS$$PROMPT_ALL
47 0047 1 should be called. SBL 15-Dec-1982
48 0048 1 --
```

PASSLOOK_AHEAD
1-002

Resolve lazy-lookahead
Declarations

J 7
16-Sep-1984 01:44:39
14-Sep-1984 12:51:39

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASLOOKAH.B32;1

Page 2
(2)

```
: 50      0049 1 %SBTTL 'Declarations'
: 51      0050 1
: 52      0051 1 PROLOGUE DEFINITIONS:
: 53      0052 1
: 54      0053 1
: 55      0054 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 56      0118 1
: 57      0119 1
: 58      0120 1 TABLE OF CONTENTS:
: 59      0121 1
: 60      0122 1
: 61      0123 1 FORWARD ROUTINE
: 62      0124 1     PASSLOOK_AHEAD: CALL_SAVEALL NOVALUE,      ! Resolve lazy-lookahead
: 63      0125 1     PASS$LOOK_AHEAD: JSB_LOOK_AHEAD NOVALUE;   ! Internally callable
: 64      0126 1
: 65      0127 1
: 66      0128 1 MACROS:
: 67      0129 1
: 68      0130 1     NONE
: 69      0131 1
: 70      0132 1 EQUATED SYMBOLS:
: 71      0133 1
: 72      0134 1     NONE
: 73      0135 1
: 74      0136 1 FIELDS:
: 75      0137 1
: 76      0138 1     NONE
: 77      0139 1
: 78      0140 1 OWN STORAGE:
: 79      0141 1
: 80      0142 1     NONE
: 81      0143 1
```



```

83 0144 1 %SBTTL 'PASSLOOK_AHEAD - Resolve lazy lookahead'
84 0145 1 GLOBAL ROUTINE PASSLOOK_AHEAD (      ! Resolve lazy lookahead
85 0146 1      PFV: REF $PASSPFV_FILE_VARIABLE ! File variable
86 0147 1      ): CALL_SAVEALL NOVALUE =
87 0148 1
88 0149 1
89 0150 1 ++
90 0151 1 FUNCTIONAL DESCRIPTION:
91 0152 1      PASSLOOKAHEAD is to be called if a textfile file buffer is
92 0153 1      undefined (PFV$V_DFB clear) and the caller wishes to access
93 0154 1      the file buffer. See PASS$LOOK_AHEAD for more information.
94 0155 1
95 0156 1 CALLING SEQUENCE:
96 0157 1
97 0158 1      CALL PASSLOOK_AHEAD (PFV.mr.r)
98 0159 1
99 0160 1 FORMAL PARAMETERS:
100 0161 1
101 0162 1      PFV      - The Pascal File Variable (PFV) passed by reference.
102 0163 1      The structure of the PFV is defined in PASPFV.REQ.
103 0164 1
104 0165 1 IMPLICIT INPUTS:
105 0166 1
106 0167 1      NONE
107 0168 1
108 0169 1 IMPLICIT OUTPUTS:
109 0170 1
110 0171 1      NONE
111 0172 1
112 0173 1 ROUTINE VALUE:
113 0174 1
114 0175 1      NONE
115 0176 1
116 0177 1 SIDE EFFECTS:
117 0178 1
118 0179 1      See PASS$LOOK_AHEAD
119 0180 1
120 0181 1 SIGNALLED ERRORS:
121 0182 1
122 0183 1
123 0184 1 --
124 0185 1
125 0186 2 BEGIN
126 0187 2
127 0188 2 LOCAL
128 0189 2      FCB: REF $PASSFCB_CONTROL_BLOCK,      ! File control block
129 0190 2      PFV_ADDR: VOLATILE,                  ! Enable argument
130 0191 2      UNWIND_ACT: VOLATILE;                ! Enable argument
131 0192 2
132 0193 2 ENABLE
133 0194 2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT);
134 0195 2
135 0196 2 ++
136 0197 2      Set enable argument for PFV address.
137 0198 2
138 0199 2
139 0200 2      PFV_ADDR = PFV [PFV$R_PFV];      ! Set PFV address
```

PASSLOOK_AHEAD
1-002

Resolve lazy-lookahead
PASSLOOK_AHEAD - Resolve lazy lookahead

L 7
16-Sep-1984 01:44:39
14-Sep-1984 12:51:39

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASLOOKAH.B32;1

Page 4
(3)

```

140 0201 2
141 0202 2
142 0203 2
143 0204 2
144 0205 2
145 0206 2
146 0207 2
147 0208 2
148 0209 2
149 0210 2
150 0211 2
151 0212 2
152 0213 2
153 0214 2
154 0215 2
155 0216 2
156 0217 2
157 0218 2
158 0219 2
159 0220 2
160 0221 2
161 0222 2
162 0223 2
163 0224 2
164 0225 2
165 0226 2
166 0227 2
167 0228 1

!+
!- Validate and lock PFV

PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);

!+
!- Set unwind action to unlock file.

UNWIND_ACT = PASS$K_UNWIND_UNLOCK;

!+
!- Call PASS$LOOK_AHEAD to do the work.

PASS$LOOK_AHEAD (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);

!+
!- Unlock file and return.

PFV [PFV$V_LOCK] = 0;

RETURN;

END;
```

! End of routine PASS\$LOOK_AHEAD

```

.TITLE PASSLOOK_AHEAD Resolve lazy-lookahead
.IDENT \1-002\

.EXTRN PASSLOOK_AHEAD, PASS$LOOK_AHEAD
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VALIDATE_PFV
```

.PSECT _PASS\$CODE,NOWRT, SHR, PIC,2

```

00CF 00000
5E 04 C2 00002
7E D4 00005
04 04 AE D4 00007
6D 001B CF DE 0000A
56 04 AC D0 0000F
04 AE 56 D0 00013
00000000G 00 16 00017
6E 01 D0 0001D
0000V 30 00020
07 A6 80 8F 8A 00023
04 00028
0000 00029 1$:
50 08 AC D0 0002B
50 04 A0 D0 0002F
F8 A0 9F 00033
FC A0 9F 00036
02 DD 00039
```

```

.ENTRY PASSLOOK_AHEAD, Save R0,R1,R2,R3,R6,R7
SUBL2 #4, SP
CLRL UNWIND_ACT
CLRL PFV_ADDR
MOVAL 1$, (FP)
MOVL PFV, R6
MOVL R6, PFV_ADDR
JSB PASS$VALIDATE_PFV
MOVL #1, UNWIND_ACT
BSBW PASS$LOOK_AHEAD
BICB2 #128, 7(R6)
RET
.WORD Save nothing
MOVL 8(AP), R0
MOVL 4(R0), R0
PUSHAB UNWIND_ACT
PUSHAB PFV_ADDR
PUSHL #2
```

```

: 0145
: 0186
: 0200
: 0206
: 0212
: 0218
: 0224
: 0228
: 0186
:
```


PASSLOOK_AHEAD Resolve lazy-lookahead
1-002 PASSLOOK_AHEAD - Resolve lazy lookahead

M 7
16-Sep-1984 01:44:39 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:39 [PASRTL.SRC]PASLOOKAH.B32;1

Page 5
(3)

			5E	DD	0003B	PUSHL	SP	
			AC	7D	0003D	MOVQ	4(AP), -(SP)	
			03	FB	00041	CALLS	#3, PASS\$IO_HANDLER	
			04	00	0048	RET		

00000000G 7E 00 04

; Routine Size: 73 bytes, Routine Base: _PASS\$CODE + 0000

; 168 0229 1
; 169 0230 1 !<BLF/PAGE>

```

171 0231 1 %SBTTL 'PASS$LOOK_AHEAD - Internally callable procedure'
172 0232 1 GLOBAL ROUTINE PASS$LOOK_AHEAD (      | Resolve lazy lookahead
173 0233 1      PFV: REF $PASS$PFV FILE VARIABLE, | File variable
174 0234 1      IN_FCB: REF $PASS$FCB CONTROL_BLOCK; | Control block
175 0235 1      FCB: REF $PASS$FCB CONTROL_BLOCK | Output FCB
176 0236 1      ): JSB_LOOK_AHEAD NOV$VALUE =
177 0237 1
178 0238 1 ++
179 0239 1 FUNCTIONAL DESCRIPTION:
180 0240 1
181 0241 1 PASS$LOOK_AHEAD is to be called if a textfile file buffer is
182 0242 1 undefined (PFV$V_DFB clear) and the caller wishes to access
183 0243 1 the file buffer. The following steps are taken.
184 0244 1
185 0245 1 1. If the file is open, and FCB$V_LAZY is clear,
186 0246 1 return.
187 0247 1 2. If the file is not open and is not INPUT or OUTPUT,
188 0248 1 return.
189 0249 1 3. If the file is not open, and is INPUT or OUTPUT,
190 0250 1 open it.
191 0251 1 4. If FCB$V_LAZY is set, then:
192 0252 1 - If file is enabled for prompting, output
193 0253 1 partial lines on all enabled files.
194 0254 1 - Do the $GET from the file.
195 0255 1 - Clear FCB$V_LAZY, set PFV$V_EOLN, PFV$V_EOF_DEFINED,
196 0256 1 PFV$V_DFB, PFV$V_VALID as appropriate.
197 0257 1
198 0258 1 CALLING SEQUENCE:
199 0259 1
200 0260 1 JSB_LOOK_AHEAD PASS$LOOK_AHEAD (PFV.mr.r, IN_FCB.mr.r;
201 0261 1 FCB.mr.r)
202 0262 1
203 0263 1 FORMAL PARAMETERS:
204 0264 1
205 0265 1 PFV - The Pascal File Variable (PFV) for the file.
206 0266 1
207 0267 1 IN_FCB - The File Control Block (FCB) for the file.
208 0268 1
209 0269 1 FCB - The result FCB for the file.
210 0270 1
211 0271 1 IMPLICIT INPUTS:
212 0272 1
213 0273 1 It is assumed that the caller has verified the PFV and has locked
214 0274 1 it.
215 0275 1
216 0276 1 IMPLICIT OUTPUTS:
217 0277 1
218 0278 1 NONE
219 0279 1
220 0280 1 COMPLETION STATUS:
221 0281 1
222 0282 1 NONE
223 0283 1
224 0284 1 SIDE EFFECTS:
225 0285 1
226 0286 1 NONE
227 0287 1

```



```
228 0288 1 ! SIGNALLED ERRORS:
229 0289 1 !
230 0290 1 !
231 0291 1 !
232 0292 1 !
233 0293 2 BEGIN
234 0294 2 LOCAL
235 0295 2 STATUS;
236 0296 2
237 0297 2
238 0298 2 BIND
239 0299 2 RAB = FCB: REF BLOCK [, BYTE]; ! RAB is also FCB address
240 0300 2
241 0301 2 BUILTIN
242 0302 2 TESTBITSC;
243 0303 2
244 0304 2 FCB = IN_FCB [FCB$R_FCB]; ! Set output FCB
245 0305 2
246 0306 2 !+
247 0307 2 See if the file is open. If not, but is INPUT or OUTPUT, open it.
248 0308 2 Otherwise, return.
249 0309 2 !-
250 0310 2
251 0311 2 IF NOT .PFV [PFV$V_OPEN] ! Not open
252 0312 2 THEN
253 0313 3 BEGIN
254 0314 3 PASS$OPEN_IMPLICIT (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
255 0315 3 IF NOT .PFV [PFV$V_OPEN] ! Still not open?
256 0316 3 THEN
257 0317 3 RETURN;
258 0318 2 END;
259 0319 2
260 0320 2 !+
261 0321 2 Indicate that buffer access is valid.
262 0322 2 !-
263 0323 2
264 0324 2 PFV [PFV$V_EOF_DEFINED] = 1;
265 0325 2 PFV [PFV$V_VALID] = 1;
266 0326 2
267 0327 2 !+
268 0328 2 Is lazy lookahead in progress on this file?
269 0329 2 !-
270 0330 2
271 0331 2 IF TESTBITSC (FCB [FCB$V_LAZY])
272 0332 2 THEN
273 0333 3 BEGIN
274 0334 3
275 0335 3 !+
276 0336 3 If EOF is true or if this is a string file (READV),
277 0337 3 signal "GET after end-of-file".
278 0338 3 !-
279 0339 3
280 0340 3 IF .FCB [FCB$V_EOF] OR .FCB [FCB$V_STRING]
281 0341 3 THEN
282 0342 3 $PASSIO_ERROR (PASS$_GETAFTEOF,0);
283 0343 3
284 0344 3 !+
```

```
285 0345 3      ! Lazy lookahead is in progress.
286 0346 3      ! If this file can initiate prompting, prompt on all enabled files.
287 0347 3      ! Then do the $GET.
288 0348 3      !-
289 0349 3
290 0350 3      IF .FCB [FCB$V_INITIATE_PROMPT]
291 0351 3      THEN
292 0352 3          PASS$PROMPT_ALL ();
293 0353 3
294 0354 3      STATUS = $PASS$RMS_OP ($GET (RAB=.RAB));
295 0355 3
296 0356 3      !+
297 0357 3      ! Set up record pointers and move first character to user's
298 0358 3      ! file buffer. Do this before we check for errors.
299 0359 3      !-
300 0360 3
301 0361 3      FCB [FCB$A_RECORD_BEG] = .RAB [RAB$L_RBF];
302 0362 3      FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_BEG];
303 0363 3      FCB [FCB$L_RECORD_LEN] = .RAB [RAB$W_RSZ];
304 0364 3      FCB [FCB$A_RECORD_END] = .FCB [FCB$A_RECORD_BEG] + .FCB [FCB$L_RECORD_LEN];
305 0365 3      IF .RAB [RAB$W_RSZ] EQL 0
306 0366 3      THEN
307 0367 4          BEGIN
308 0368 4              PFV [PFV$V_EOLN] = 1;
309 0369 4              CH$WCHAR (%C' ', .PFV [PFV$A_BUFFER]);
310 0370 4          END
311 0371 3      ELSE
312 0372 4          BEGIN
313 0373 4              PFV [PFV$V_EOLN] = 0;
314 0374 4              CH$WCHAR (.FCB [FCB$A_RECORD_CUR], .PFV [PFV$A_BUFFER]);
315 0375 4          END;
316 0376 3
317 0377 3      IF NOT .STATUS
318 0378 3      THEN
319 0379 4          BEGIN
320 0380 4              !+
321 0381 4              ! $GET failed. If RMS$_EOF, simply set EOF. Otherwise
322 0382 4              ! signal an error.
323 0383 4              !-
324 0384 4              IF .RAB [RAB$L_STS] NEQ RMS$_EOF
325 0385 4              THEN
326 0386 4                  $PASS$IO_ERROR (PASS$_ERRDURGET); ! Error during GET
327 0387 4                  FCB [FCB$V_EOF] = 1;
328 0388 4                  PFV [PFV$V_DFB] = 0;
329 0389 4                  PFV [PFV$V_EOLN] = 0;
330 0390 4                  END
331 0391 3              ELSE
332 0392 4                  BEGIN
333 0393 4                      FCB [FCB$V_EOF] = 0;
334 0394 4                      PFV [PFV$V_DFB] = 1;
335 0395 4                      FCB [FCB$L_RECORD_NUMBER] = .FCB [FCB$L_RECORD_NUMBER] + 1;
336 0396 4                  END;
337 0397 3
338 0398 2      END;
339 0399 2
340 0400 2
341 0401 2      RETURN;
```


: 342
: 3430402 2
0403 1 END;

! End of routine PAS\$\$LOOK_AHEAD

.EXTRN PAS\$\$OPEN_IMPLICIT
.EXTRN PAS\$\$SIGNAL, PASSK_GETAFTEOF
.EXTRN PAS\$\$PROMPT_ALL
.EXTRN SY\$\$GET, PASSK_ERRDURGET

	53	04	A6	9E	00000	PAS\$\$LOOK_AHEAD::		
						MOVAB	4(PFV), R3	: 0311
0B	63		1D	E0	00004	BBS	#29, (R3), 1\$	: 0314
		00000000G	00	16	00008	JSB	PAS\$\$OPEN_IMPLICIT	: 0315
01	63		1D	E0	0000E	BBS	#29, (R3), 1\$	
				05	00012	RSB		
	02	A3	05	88	00013	BISB2	#5, 2(R3)	: 0325
		52	FC	A7	9E	MOVAB	-4(FCB), R2	: 0331
01	62		0A	E4	0001B	BBSC	#10, (R2), 2\$	
				05	0001F	RSB		
04	62		0D	E0	00020	BBS	#13, (R2), 3\$	: 0340
0E	62		14	E1	00024	BBC	#20, (R2), 4\$	
			7E	D4	00028	CLRL	-(SP)	: 0342
	7E	00G	8F	9A	0002A	MOVZBL	#PASSK_GETAFTEOF, -(SP)	
	00000000G	00	02	FB	0002E	CALLS	#2, PAS\$\$SIGNAL	
				05	00035	RSB		
07	62		19	E1	00036	BBC	#25, (R2), 5\$	: 0350
	00000000G	00	00	FB	0003A	CALLS	#0, PAS\$\$PROMPT_ALL	: 0352
			57	DD	00041	PUSHL	FCB	: 0354
	00000000G	00	01	FB	00043	CALLS	#1, SY\$\$GET	
	0D		50	E8	0004A	BLBS	\$\$STATUS, 6\$	
	0001825A	8F	50	D1	0004D	CMPL	\$\$STATUS, #98906	
			04	12	00054	BNEQ	6\$	
	E7	03	A2	E8	00056	BLBS	3(R2), 5\$	
	E8	28	A7	D0	0005A	MOVL	40(FCB), -24(FCB)	: 0361
	EC	E8	A7	D0	0005F	MOVL	-24(FCB), -20(FCB)	: 0362
	F4	22	A7	3C	00064	MOVZWL	34(FCB), -12(FCB)	: 0363
F0	A7	F4	A7	C1	00069	ADDL3	-12(FCB), -24(FCB), -16(FCB)	: 0364
		22	A7	B5	00070	TSTW	34(FCB)	: 0365
			0A	12	00073	BNEQ	7\$	
	02	A3	08	88	00075	BISB2	#8, 2(R3)	: 0368
	00	B6	20	90	00079	MOVB	#32, a0(PFV)	: 0369
			09	11	0007D	BRB	8\$	: 0365
	02	A3	08	8A	0007F	BICB2	#8, 2(R3)	: 0373
	00	B6	EC	B7	90	MOVB	a-20(FCB), a0(PFV)	: 0374
	1F		50	E8	00088	BLBS	STATUS, 10\$	: 0377
	0001827A	8F	08	A7	D1	CMPL	8(FCB), #98938	: 0384
			0C	13	00093	BEQL	9\$	
	7E	00G	8F	9A	00095	MOVZBL	#PASSK_ERRDURGET, -(SP)	: 0386
	00000000G	00	01	FB	00099	CALLS	#1, PAS\$\$SIGNAL	
				05	000A0	RSB		
	01	A2	20	88	000A1	BISB2	#32, 1(R2)	: 0387
	02	A3	0A	8A	000A5	BICB2	#10, 2(R3)	: 0389
				05	000A9	RSB		: 0377
	01	A2	20	8A	000AA	BICB2	#32, 1(R2)	: 0393
	02	A3	02	88	000AE	BISB2	#2, 2(R3)	: 0394
		C8	A7	D6	000B2	INCL	-56(FCB)	: 0395

: Routine Size: 182 bytes, Routine Base: _PASS\$CODE + 0049

```

: 344      0404 1
: 345      0405 1 !<BLF/PAGE>
: 346      0406 1 END
: 347      0407 1
: 348      0408 0 ELUDOM

```

! End of module PASS\$LOOK_AHEAD

PSECT SUMMARY		
Name	Bytes	Attributes
_PASS\$CODE	255	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics					
File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	9	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	96	22	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASLOOKAH/OBJ=OBJ\$:PASLOOKAH MSRC\$:PASLOOKAH/UPDATE=(ENH\$:PASLOOKAH)

```

: Size:      255 code + 0 data bytes
: Run Time:   00:08.0
: Elapsed Time: 00:28.6
: Lines/CPU Min: 3079
: Lexemes/CPU-Min: 17094
: Memory Used: 112 pages
: Compilation Complete

```


0295 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY